PENDING CLAIMS

1-58 (Cancelled)

59. (New) An application software program, comprising:

an object-oriented, verifiable, type-safe and pointer-safe sequence of instructions residing on a computer-readable medium, said instructions comprising operation codes and operands, said program operable to be loaded to and executed by a resource-constrained device, said instructions previously converted from at least one class file, said conversion transforming at least one reference to a constant pool to inline data in said instructions.

60. (New) The software program of claim 59 wherein one or more of said references to said constant pool are transformed into inline data in operands in one or more of said instructions.

61. (New) The software program of claim 59 wherein one or more of said references to said constant pool are transformed into inline data in operation codes in one or more of said instructions.

62. (New) The software program of claim 59 wherein said resource-constrained device is based on a 16-bit processor architecture.

63. (New) The software program of claim 59 wherein said resource-constrained device is based on an 8-bit processor architecture.

64. (New) The software program of claim 59 wherein said resource-constrained device comprises a random access memory with a capacity of no more than about 64 kilo-bytes.

65. (New) The software program of claim 59 wherein said resource-constrained device comprises a random access memory with a capacity of no more than about 4 kilo-bytes.

66. (New) The software program of claim 59 wherein said instructions are configured for execution by a virtual machine running on a microprocessor residing on said resource-constrained device.

67. (New) The software program of claim 59 wherein said resource-constrained device comprises a smart card.

68. (New) The software program of claim 59 wherein said resource-constrained device comprises an application-specific integrated circuit (ASIC).

69. (New) An application software program, comprising:

an object-oriented, verifiable, type-safe and pointer-safe sequence of instructions residing on a computer-readable medium, said instructions comprising operation codes and operands, said program operable to be loaded to and executed by a resource-constrained

device, said instructions previously converted from at least one class file, said instructions comprising at least one composite instruction for performing an operation on a current object.

70. (New) The software program of claim 69 wherein said resource-constrained device is based on a 16-bit processor architecture.

71. (New) The software program of claim 69 wherein said resource-constrained device is based on an 8-bit processor architecture.

72. (New) The software program of claim 69 wherein said resource-constrained device comprises a random access memory with a capacity of no more than about 64 kilo-bytes.

73. (New) The software program of claim 69 wherein said resource-constrained device comprises a random access memory with a capacity of no more than about 4 kilo-bytes.

74. (New) The software program of claim 69 wherein said instructions are configured for execution by a virtual machine running on a microprocessor residing on said resource-constrained device.

75. (New) The software program of claim 69 wherein said resource-constrained device comprises a smart card.

76. (New) The software program of claim 69 wherein said resource-constrained device comprises an application-specific integrated circuit (ASIC).

77. (New) A resource-constrained device comprising:

a memory for storing an application software program comprising an object-oriented, verifiable, type-safe and pointer-safe sequence of instructions, said instructions comprising operation codes and operands, said instructions previously converted from at least one class file, said conversion transforming at least one reference to a constant pool to inline data in said instructions; and

a virtual machine implemented on a microprocessor, said virtual machine configured to execute said sequence of instructions.

78. (New) The resource-constrained device of claim 77 wherein one or more of said references to said constant pool are transformed into inline data in operands in one or more of said instructions.

79. (New) The resource-constrained device of claim 77 wherein one or more of said references to said constant pool are transformed into inline data in operation codes in one or more of said instructions.

80. (New) The resource-constrained device of claim 77 wherein said resource-constrained device is based on a 16-bit processor architecture.

81. (New) The resource-constrained device of claim 77 wherein said resource-constrained device is based on an 8-bit processor architecture.

82. (New) The resource-constrained device of claim 77 wherein said resource-constrained device comprises a random access memory with a capacity of no more than about 64 kilo-bytes.

83. (New) The resource-constrained device of claim 77 wherein said resource-constrained device comprises a random access memory with a capacity of no more than about 4 kilo-bytes.

84. (New) The resource-constrained device of claim 77 wherein said resource-constrained device comprises a smart card.

85. (New) The resource-constrained device of claim 77 wherein said resource-constrained device comprises an application-specific integrated circuit (ASIC).

86. (New) The resource-constrained device of claim 77 wherein said resource-constrained device comprises a Java Card™ technology-enabled smart card.

87. (New) A resource-constrained device comprising:

a memory for storing an application software program comprising an object-oriented, verifiable, type-safe and pointer-safe sequence of instructions, said instructions

comprising operation codes and operands, said instructions previously converted from at least one class file, said instructions comprising at least one composite instruction for performing an operation on a current object; and

a virtual machine implemented on a microprocessor, said virtual machine configured to execute said sequence of instructions.

88. (New) The resource-constrained device of claim 87 wherein said resource-constrained device is based on a 16-bit processor architecture.

89. (New) The resource-constrained device of claim 87 wherein said resource-constrained device is based on an 8-bit processor architecture.

90. (New) The resource-constrained device of claim 87 wherein said resource-constrained device comprises a random access memory with a capacity of no more than about 64 kilo-bytes.

91. (New) The resource-constrained device of claim 87 wherein said resource-constrained device comprises a random access memory with a capacity of no more than about 4 kilo-bytes.

92. (New) The resource-constrained device of claim 87 wherein said resource-constrained device comprises a smart card.

93. (New) The resource-constrained device of claim 87 wherein said resource-constrained device comprises an application-specific integrated circuit (ASIC).

94. (New) The resource-constrained device of claim 87 wherein said resource-constrained device comprises a Java Card™ technology-enabled smart card.

95. (New) A method for using an application software program including an object-oriented, verifiable, type-safe and pointer-safe sequence of instructions, the method comprising:

receiving said software program in a resource-constrained device having a memory, said instructions comprising operation codes and operands, said instructions previously converted from at least one class file, said conversion transforming at least one reference to a constant pool to inline data in said instructions; and

executing said sequence of instructions on said resource-constrained device.

96. (New) The method of claim 95, further comprising accessing said software program over a computer network prior to downloading said program onto said resource-constrained device.

97. (New) The method of claim 95, further comprising storing said sequence of instructions on said resource-constrained device.

98. (New) The method of claim 95, further comprising accessing said software program over the Internet prior to downloading said program onto said resource-constrained device.

99. (New) The method of claim 95, further comprising transforming constant pool indices that appear in the received set of instructions to corresponding data values.

100. (New) The method of claim 95 wherein one or more of said references to said constant pool are transformed into inline data in operands in one or more of said instructions.

101. (New) The method of claim 95 wherein one or more of said references to said constant pool are transformed into inline data in operation codes in one or more of said instructions.

102. (New) The method of claim 95 wherein said resource-constrained device is based on a 16-bit processor architecture.

103. (New) The method of claim 95 wherein said resource-constrained device is based on an 8-bit processor architecture.

104. (New) The method of claim 95 wherein said memory comprises a random access memory with a capacity of no more than about 64 kilo-bytes.

105. (New) The method of claim 95 wherein said memory comprises a random access memory with a capacity of no more than about 4 kilo-bytes.

106. (New) The method of claim 95 wherein said instructions are configured for execution by a virtual machine running on a microprocessor residing on said resource-constrained device.

107. (New) The method of claim 95 wherein said resource-constrained device comprises a smart card.

108. (New) The method of claim 95 wherein said resource-constrained device comprises an application-specific integrated circuit (ASIC).

109. (New) A method for using an application software program including an object-oriented, verifiable, type-safe and pointer-safe sequence of instructions, the method comprising:

receiving said software program in a resource-constrained device having a memory, said instructions comprising operation codes and operands, said instructions previously converted from at least one class file, said instructions comprising at least one composite instruction for performing an operation on a current object; and

executing said sequence of instructions on said resource-constrained device.

110. (New) The method of claim 109, further comprising accessing said software program over a computer network prior to downloading said program onto said resource-constrained device.

111. (New) The method of claim 109, further comprising storing said sequence of instructions on said resource-constrained device.

112. (New) The method of claim 109, further comprising accessing said software program over the Internet prior to downloading said program onto said resource-constrained device.

113. (New) The method of claim 109, further comprising transforming constant pool indices that appear in the received set of instructions to corresponding data values.

114. (New) The method of claim 109 wherein one or more of said references to said constant pool are transformed into inline data in operands in one or more of said instructions.

115. (New) The method of claim 109 wherein one or more of said references to said constant pool are transformed into inline data in operation codes in one or more of said instructions.

116. (New) The method of claim 109 wherein said resource-constrained device is based on a 16-bit processor architecture.

117. (New) The method of claim 109 wherein said resource-constrained device is based on an 8-bit processor architecture.

118. (New) The method of claim 109 wherein said memory comprises a random access memory with a capacity of no more than about 64 kilo-bytes.

119. (New) The method of claim 109 wherein said memory comprises a random access memory with a capacity of no more than about 4 kilo-bytes.

120. (New) The method of claim 109 wherein said instructions are configured for execution by a virtual machine running on a microprocessor residing on said resource-constrained device.

121. (New) The method of claim 109 wherein said resource-constrained device comprises a smart card.

122. (New) The method of claim 109 wherein said resource-constrained device comprises an application-specific integrated circuit (ASIC).

123. (New) An apparatus for using an application software program including an object-oriented, verifiable, type-safe and pointer-safe sequence of instructions, the apparatus comprising:

means for receiving said software program in a resource-constrained device having a memory, said instructions comprising operation codes and operands, said instructions previously converted from at least one class file, said conversion transforming at least one reference to a constant pool to inline data in said instructions; and

means for executing said sequence of instructions on said resource-constrained device.

124. (New) The apparatus of claim 123, further comprising means for accessing said software program over a computer network prior to downloading said program onto said resource-constrained device.

125. (New) The apparatus of claim 123, further comprising means for storing said sequence of instructions on said resource-constrained device.

126. (New) The apparatus of claim 123, further comprising means for accessing said software program over the Internet prior to downloading said program onto said resource-constrained device.

127. (New) The apparatus of claim 123, further comprising means for transforming constant pool indices that appear in the received set of instructions to corresponding data values.

128. (New) The apparatus of claim 123 wherein one or more of said references to said constant pool are transformed into inline data in operands in one or more of said instructions.

129. (New) The apparatus of claim 123 wherein one or more of said references to said constant pool are transformed into inline data in operation codes in one or more of said instructions.

130. (New) The apparatus of claim 123 wherein said resource-constrained device is based on a 16-bit processor architecture.

131. (New) The apparatus of claim 123 wherein said resource-constrained device is based on an 8-bit processor architecture.

132. (New) The apparatus of claim 123 wherein said memory comprises a random access memory with a capacity of no more than about 64 kilo-bytes.

133. (New) The apparatus of claim 123 wherein said memory comprises a random access memory with a capacity of no more than about 4 kilo-bytes.

134. (New) The apparatus of claim 123 wherein said instructions are configured for execution by a virtual machine running on a microprocessor residing on said resource-constrained device.

135. (New) The apparatus of claim 123 wherein said resource-constrained device comprises a smart card.

136. (New) The apparatus of claim 123 wherein said resource-constrained device comprises an application-specific integrated circuit (ASIC).

137. (New) An apparatus for using an application software program including an object-oriented, verifiable, type-safe and pointer-safe sequence of instructions, the apparatus comprising:

means for receiving said software program in a resource-constrained device having a memory, said instructions comprising operation codes and operands, said instructions previously converted from at least one class file, said instructions comprising at least one composite instruction for performing an operation on a current object; and

means for executing said sequence of instructions on said resource-constrained device.

138. (New) The apparatus of claim 137, further comprising means for accessing said software program over a computer network prior to downloading said program onto said resource-constrained device.

139. (New) The apparatus of claim 137, further comprising means for storing said sequence of instructions on said resource-constrained device.

140. (New) The apparatus of claim 137, further comprising means for accessing said software program over the Internet prior to downloading said program onto said resource-constrained device.

141. (New) The apparatus of claim 137, further comprising means for transforming constant pool indices that appear in the received set of instructions to corresponding data values.

142. (New) The apparatus of claim 137 wherein one or more of said references to said constant pool are transformed into inline data in operands in one or more of said instructions.

143. (New) The apparatus of claim 137 wherein one or more of said references to said constant pool are transformed into inline data in operation codes in one or more of said instructions.

144. (New) The apparatus of claim 137 wherein said resource-constrained device is based on a 16-bit processor architecture.

145. (New) The apparatus of claim 137 wherein said resource-constrained device is based on an 8-bit processor architecture.

146. (New) The apparatus of claim 137 wherein said memory comprises a random access memory with a capacity of no more than about 64 kilo-bytes.

147. (New) The apparatus of claim 137 wherein said memory comprises a random access memory with a capacity of no more than about 4 kilo-bytes.

148. (New) The apparatus of claim 137 wherein said instructions are configured for execution by a virtual machine running on a microprocessor residing on said resource-constrained device.

149. (New) The apparatus of claim 137 wherein said resource-constrained device comprises a smart card.

150. (New) The apparatus of claim 137 wherein said resource-constrained device comprises an application-specific integrated circuit (ASIC).